



Usando Loop para carga de datos

Transcripción

[00:00] Vamos a volver aquí a nuestro código, ya con ese concepto repasado, y lo que necesitamos es, si yo definí cuatro valores para mi arreglo, crear una iteración que itere, que pase cuatro veces consultando: ¿El input del usuario es igual a secreto posición 1? ¿Sí o no? ¿El input del usuario es igual a la posición de secreto 2? Y así sucesivamente.

[00:29] Entonces, para eso vamos a hacer uso de un concepto que ya aprendimos en las anteriores aulas que son las iteraciones. Vamos a usar un for, y en este for vamos ya a dejar nuestra estructura lista. Aquí adentro vamos a colocar la condición. Y recordando que el for tiene cuatro partes: una primera, una segunda y una tercera.

[01:07] La primer parte es el inicializador de la variable. Entonces, en un for necesito un contador, pero, en este caso, a nuestro contador para tener una variable más explícita, que explique lo que está haciendo, vamos a llamarlo de posición, que va a ser la posición de nuestro arreglo que hemos creado. Entonces, var posición igual a 0.

[01:33] Aquí posición es la condición, la segunda parte de nuestro for. En la condición tenemos cuatro elementos. Entonces, le voy a decir posición menor a 4. Aquí tengo dos opciones. Como posición está empezando en 0. Yo le puedo decir "posición menor o igual a 3" o le puedo decir "posición menor a 4".

[02:04] Porque si es posición menor a 4 empieza con 0 en la segunda iteración se vuelve 1, continúa pasando, en la tercera iteración se vuelve 2, continúa

pasando, en la cuarta iteración se vuelve 3, continúa pasando, y en la quinta iteración posición pasa a ser 4. Ya no pasa por esta condición, porque ya no es menor que 4, en el caso es igual.

[02:35] Y la tercer parte de nuestro for es el sumador de la variable.

Posición++. Entonces, nuestra variable va a entrar aquí cuatro veces. Lo que necesitamos aquí es modificar nuestra variable para secretos, ya no es secreto, y aquí le pasamos el índice que es nuestra variable posición.

[02:57] ¿Entonces qué va a hacer nuestro programa? Voy a guardarlo. Nuestro programa va a entrar, vamos a llamar a la función verificar nuevamente y va a hacer una iteración de cuatro veces aquí. Entonces va a preguntar si es el valor, nos va a decir si acertó o no, va a entrar de nuevo, nos va a decir si acertó o no y así sucesivamente.

[03:19] Vamos a verlo. Voy ahora a cerrar esta parte, actualizo aquí y pasamos un valor 2, vamos a errar primero. "Usted erró", "Usted erró", "Usted erró" y listo. Me lanza cuatro mensajes porque son cuatro iteraciones. Si yo acierto el número 5, en la primera erré, en la segunda acerté, erré y erró.

[03:47] ¿Qué necesito alterar para no tener ese problema de que me está lanzando varios mensajes innecesarios? Este else ya no lo tengo aquí. Y lo vamos de momento, sin el else, colocar aquí abajo, solo momentáneamente. Entonces, aquí era "Usted erró".

[04:17] Nuestro programa lo que va a hacer es va a entrar aquí, va a preguntar en cuatro veces si acertamos. Si acertamos, nos va a decir que acertó. Si no, nos va a decir que erró. Vamos aquí, actualizamos aquí y comenzamos. Un número 5, que es parte. "Usted acertó". Ahora aquí está uno de los problemas.

[04:49] Fíjense que entra en el for y vuelve, después de las cuatro iteraciones, independientemente de si yo acerté o erré, me va a decir que yo erré. ¿Cómo resuelvo este dilema? Voy a crear aquí dentro de la misma función una nueva

variable que la voy a llamar encontrado y va a ser booleano, va a ser o falso o verdadero y lo voy a definir como falso.

[05:26] Entonces, una vez que encontré, o sea, si acerté, lo voy a alterar para true, y aquí, al final de mi for, dentro de la función, voy a hacer un if: si encontrado fuera igual que false, que nos presente el alert. Perfecto. Aquí, solo un detalle más que es este de aquí. Vamos a rodarlo para mostrarles ese problema.

[06:25] Volvemos aquí y actualizamos. Entonces, voy a lanzar un 3. "Usted acertó". Listo, ya no entra. Voy a lanzar un 4. "Usted erró". Ya no entra. Entonces, el programa está funcionando correctamente, perfecto. Solo que aquí hay un detalle que tenemos que tener cuidado cuando usamos los for.

[06:52] Fíjense que, si yo coloqué 3, significa que en la primer iteración ya debería haber acertado, no necesito entrar a la segunda ni a la tercera ni a la cuarta. Entonces, para hacer que mi programa sea más eficiente, aquí se acostumbra colocar un break que sea cuando entre a la condición verdadera, cuando acierte el valor, ya ni necesito verificar los otros valores. ¿Están de acuerdo?

[07:26] Porque imagínense. Ahorita tengo cuatro posiciones, son cuatro iteraciones, nuestro programa lo hace superrápido, en cuestión de milisegundos. Pero si nuestra iteración fuera de millones, si nuestro vector, nuestro arreglo tuviera millones de valores, la población de una ciudad, ahí ya no van a ser microsegundos lo que va a demorar. ¿Correcto?

[07:50] Entonces, si yo ya obtuve resultado cierto, corto el for, ya no necesito validar y automáticamente me salgo de aquí. Vamos a guardar, volvemos acá y solo lo probamos. 2. "Usted erró". 7. "Usted acertó". 9. "Usted acertó". Entonces, cuando yo coloco el 3, es en la primera iteración que me lo muestra.

[08:20] Cuando yo aprieto 5 es en la segunda iteración, cuando hago el 7 es en la tercera y cuando hago el 9 es en la cuarta. Aquí no noto diferencia

nuevamente en la ejecución porque al ser pocos valores no demora nada, demora microsegundos en ejecutar ese for.